**C HOMEWORKS – ARRAYS**

**Omair Hamd Alla**

1. Write a Java program to sort a numeric array and a string array.

import java.util.Arrays;

public class Q1 {

public static void main(String[] args) {

int[] numericArray = {5, 2, 8, 1, 6, 3};

String[] stringArray = {"trabzon", "hatay", "izmir", "bursa", "ankara", "rize"};

System.***out***.println("Original Numeric Array: " + Arrays.*toString*(numericArray) + "\nOriginal String Array: " + Arrays.*toString*(stringArray));

*sort*(numericArray);

*sort*(stringArray);

System.***out***.println("Sorted Numeric Array: " + Arrays.*toString*(numericArray) + "Sorted String Array: " + Arrays.*toString*(stringArray));

}

private static void sort(int[] array) {

int len = array.length;

for (int i = 0; i < len - 1; i++)

for (int j = 0; j < len - i - 1; j++)

if (array[j] > array[j + 1]) {

int temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

private static void sort(String[] array) {

int len = array.length;

for (int i = 0; i < len - 1; i++)

for (int j = 0; j < len - i - 1; j++)

if (array[j].compareTo(array[j + 1]) > 0) {

String temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

}

1. Write a Java program to test if an array contains a specific value.

import java.util.Scanner;

public class Q2 {

public static void main(String[] args) {

int[] numbers = {1, 5, 8, 3, 2, 9, 7};

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Enter the num: ");

int targetValue = scanner.nextInt();

boolean found = false;

for (int i = 0; i < numbers.length; i++) {

if (numbers[i] == targetValue) {

found = true;

break;

}

}

if (found)

System.***out***.println("The array contains the value " + targetValue);

else

System.***out***.println("The array does not contain the value " + targetValue);

}

}

1. Write a Java program to find the index of an array element.

import java.util.Scanner;

public class Q3 {

public static void main(String[] args) {

int[] numbers = {1, 5, 8, 3, 2, 9, 7};

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Enter the num to search: ");

int targetValue = scanner.nextInt();

int index = -1;

boolean found = false;

for (int i = 0; i < numbers.length; i++) {

index++;

if (numbers[i] == targetValue) {

found = true;

break;

}

}

if (found)

System.***out***.println("The array contains the value " + targetValue + " At the index: " + index);

else

System.***out***.println("The array does not contain the value " + targetValue);

}

}

1. Write a Java program to remove a specific element from an array.

import java.util.Arrays;

import java.util.Scanner;

public class Q4 {

public static void main(String[] args) {

int[] numbers = {1, 5, 8, 3, 2, 9, 7};

System.***out***.println("Original Array: " + Arrays.*toString*(numbers));

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Enter the num: ");

int toRemove = scanner.nextInt();

int[] numbers2 = *removeElement*(numbers, toRemove);

System.***out***.println("Array After: " + Arrays.*toString*(numbers2));

}

public static int[] removeElement(int[] array, int toRemove) {

int index = -1;

for (int i = 0; i < array.length; i++)

if (array[i] == toRemove){

index = i;

break;}

if (index != -1) {

int newLength = array.length - 1;

int[] newArr = new int[newLength];

int j = 0;

for (int i = 0; i < array.length; i++)

if (array[i] != toRemove)

newArr[j++] = array[i];

return newArr;

} else {

System.***out***.println("Not Found");

return array;

}

}

}

1. Write a Java program to insert an element (specific position) into an array.

import java.util.Arrays;

import java.util.Scanner;

public class Q5 {

public static void main(String[] args) {

int[] numbers = {1, 5, 8, 3, 2, 9, 7};

System.***out***.println("Original Array: " + Arrays.*toString*(numbers));

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Enter the num to add: ");

int toAdd = scanner.nextInt();

System.***out***.print("Enter the index: ");

int index = scanner.nextInt();

int[] numbers2 = *addElement*(numbers, toAdd, index);

System.***out***.println("Array After: " + Arrays.*toString*(numbers2));

}

public static int[] addElement(int[] array, int toAdd, int index) {

if (index >= -1 && index < array.length) {

int newLength = array.length + 1;

int[] newArr = new int[newLength];

boolean flag = false;

for (int i = 0; i < newLength; i++) {

if (i == index) {

newArr[i] = toAdd;

flag = true;}

else if (!flag) newArr[i] = array[i];

else if (flag) newArr[i] = array[i-1];

}

return newArr;

} else {

System.***out***.println("Invalid");

return array;}

}

}

1. Write a Java program to reverse an array of integer values.

import java.util.Arrays;

public class Q6 {

public static int[] reverseArray(int[] arr) {

int[] newArr = new int[arr.length];

for (int i = arr.length - 1, j = 0; i >= 0; i--, j++) {

newArr[j] = arr[i];

}

return newArr;

}

public static void main(String[] args) {

int[] numbers = {1, 5, 8, 3, 2, 9, 7};

int[] reversedArr = *reverseArray*(numbers);

System.***out***.println("Original array: " + Arrays.*toString*(numbers));

System.***out***.println("Reversed array: " + Arrays.*toString*(reversedArr));

}

}

1. Write a Java program to find the common elements between two arrays of integers.

public class Q7 {

public static void commons(int[] numbers, int[] numbers2) {

for (int i = 0; i < numbers.length; i++) {

for (int j = 0; j < numbers2.length; j++) {

if (numbers[i] == numbers2[j]) {

System.***out***.println("Common num: " + numbers[i]);

break;

}

}

}

}

public static void main(String[] args) {

int[] numbers = {8,9,4,5,7};

int[] numbers2 = {3,2,7,4,6};

*commons*(numbers, numbers2);

}

}

1. Write a Java program to remove duplicate elements from an array.

public class Q8 {

public static void duplicates(int[] array) {

for (int i = 0; i < array.length; i++)

for (int j = i+1; j < array.length; j++)

if (array[i] == array[j]) {

System.***out***.println("Duplicated number found, Its: " + array[i]);

break;

}

}

public static void main(String[] args) {

int[] numbers = {8,2,3,7,2,5,9,3};

*duplicates*(numbers);

}

}

1. Write a Java program to test the equality of two arrays.

public class Q9 {

public static void sort(int[] array) {

for (int i = 0; i < array.length - 1; i++)

for (int j = 0; j < array.length - 1 - i; j++)

if (array[j] > array[j + 1]) {

int temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

public static void equalitiy(int[] arr1, int[] arr2) {

boolean flag = true;

for (int i = 0; i < arr1.length; i++)

if (arr1[i] != arr2[i]) {

System.***out***.println("Not equal ");

flag = false;

break;

}

if(flag) System.***out***.println("Equal");

}

public static void main(String[] args) {

int[] nums = {9,8,5,7,1};

int[] nums2 = {1,5,8,7,9};

*sort*(nums);

*sort*(nums2);

*equalitiy*(nums, nums2);

}

}

1. Write a Java program to separate even and odd numbers of an given array of integers. Put all even numbers first, and then odd numbers.

import java.util.Arrays;

public class Q10 {

public static void sort(int[] array) {

for (int i = 0; i < array.length; i++)

for (int j = i+1; j < array.length; j++)

if (array[i] % 2 != 0)

if(array[j] % 2 == 0) {

int temp = array[i];

array[i] = array[j];

array[j] = temp;

}

System.***out***.println(Arrays.*toString*(array));

}

public static void main(String[] args) {

int[] numbers = {6,1,2,4,7,9,4,5};

*sort*(numbers);

}

}

1. Write a Java program to find the k largest elements in a given array. Elements in the array can be in any order.

public class Q11 {

public static void max(int[] arr) {

int max = 0;

for (int i = 0; i < arr.length; i++)

if(max<arr[i])

max = arr[i];

System.***out***.println("Max = " + max);

}

public static void main(String[] args) {

int[] arr1 = {11 , 57 , 81 , 64 , 13 , 74};

*max*(arr1);

}

}

1. Write a Java program to find the numbers greater than the average of the numbers of a given array.

public class Q12 {

public static void higherAvg(int[] array) {

int sum = 0;

for (int i = 0; i < array.length; i++)

sum+=array[i];

int avg = sum/array.length;

System.***out***.println("average: "+ avg);

System.***out***.println("nums higher than average: ");

for(int i: array) if(i>avg) System.***out***.print(i + " , ");

}

public static void main(String[] args) {

int[] numbers = {77 , 54 , 62 , 31 , 24 , 46};

*higherAvg*(numbers);

}

}

1. Write a Java program to find the length of the longest consecutive sequence of a given array of integers.

İ got this with help from the internet all solutions were too complicated

import java.util.\*;

public class Q13 {

public static void main(String[] args) {

int arr[] = {10,21,45,22,7,2,95,19,13,14,12,11,18,16,17,100,250,21,99};

System.***out***.println("Length of the sequence we got is: "+*longestSequence*(arr));

}

private static int longestSequence(int[] arr) {

Set<Integer> set=new HashSet<Integer>();

for(int i:arr) set.add(i);

int max=0, start=0;

for(int i:arr){

int left=i-1;

int right=i+1;

int count=1;

while(set.contains(left)){

set.remove(left--);

count++;}

while(set.contains(right)){

set.remove(right++);

count++;}

if(max<count){

max=count;

start=left+1; }

}

System.***out***.println("Sequence:");

for(int i=start; i<start+max; i++) System.***out***.print(i+",");

System.***out***.println();

return max;

}

}

1. Write a Java program to divide a given array of integers into given k non-empty subsets whose sums are all equal. Return true if all sums are equal otherwise return false.

public class Q14 {

public static boolean canPartitionKSubsets(int[] nums, int k) {

int totalSum = 0;

for (int num : nums)

totalSum += num;

if (totalSum % k != 0)

return false;

int target = totalSum / k;

boolean[] visited = new boolean[nums.length];

return *canBePartition*(nums, visited, 0 , 0 , target , k);

}

private static boolean canBePartition(int[] nums, boolean[] visited, int currentSum, int currentSet, int target, int k) {

if (k == 1)

return true;

if (currentSum == target)

return *canBePartition*(nums, visited, 0, currentSet + 1, target, k - 1);

for (int i = 0; i < nums.length; i++)

if (!visited[i] && currentSum + nums[i] <= target) {

visited[i] = true;

if ( *canBePartition*(nums, visited, currentSum + nums[i], currentSet, target, k) )

return true;

visited[i] = false; // Backtrack

}

return false;

}

}

1. Write a Java program to multiply corresponding elements of two arrays of integers.

public class Q15 {

public static void main(String[] args) {

int nums1[] = {7, 3, 8, 4};

int nums2[] = {4, 1, 2, 9};

int array[] = new int[nums1.length];

for(int i = 0; i < nums1.length; i++)

array[i] = nums1[i] \* nums2[i];

System.***out***.println("Array is :");

for(int j = 0; j < nums1.length; j++)

System.***out***.print(array[j] + ", ");

}

}

1. Write a Java program to merge two given sorted array of integers and create a new sorted array.

import java.util.Arrays;

public class Q16 {

public static void sortMerge(int nums[], int nums2[]) {

int len1 = nums.length;

int len2 = nums2.length;

int array[] = new int[len1+len2];

int i = 0;

int j = 0;

int counter = 0;

while(i < len1) array[counter++] = nums[i++];

while(j < len2) array[counter++] = nums2[j++];

Arrays.*sort*(array);

System.***out***.println("final array: " + Arrays.*toString*(array));

}

public static void main(String[] args) {

int nums[] = {1,2,7,5,8,2};

int nums2[] = {5,6,10,1,7,9};

*sortMerge*(nums,nums2);

}

}

1. Write a Java program to calculate the median of an given unsorted array of integers.

import java.util.Arrays;

public class Q17 {

public static double median(int[] nums) {

Arrays.*sort*( nums);

int len = nums.length;

int middleIndex = len / 2 ;

if (len % 2 == 0)

return (nums[middleIndex - 1 ] + nums[middleIndex]) / 2 ;

else

return nums[middleIndex];

}

public static void main(String[] args) {

int[] nums = {5,8,6,4,5,1,2};

double median = *median*(nums);

System.***out***.println("Median is: " + median);

}

}

1. Write a Java program to find a number that appears only once in a given array of integers, all numbers occur twice.

public class Q18 {

public static int findUnique(int[] nums) {

int unique = 0;

for (int num : nums) {

unique ^= num;

}

return unique;

}

public static void main(String[] args) {

int[] numbers = {1, 4, 2, 1, 3, 2, 3};

int singleNumber = *findUnique*(numbers);

System.***out***.println("number appeared once is: " + singleNumber);

}

}

1. Write a Java program to remove all occurrences of a specified value in a given array of integers and return the new length of the array.

public class Q19 {

public static void main(String[] args) {

int [] nums = {6,4,7,8,9,5,1,2,1,3,6,4};

int length = nums.length;

System.***out***.println("Length of the original array: " + length);

System.***out***.print("New length after removing check: " + *check*(nums,length));

}

public static int check(int nums[], int length) {

int result = 0;

for(int i = 0; i < length; i++) {

boolean flag = false;

for(int j = 0; j < i; j++) {

if(nums[i] == nums[j]) {

flag = true;

break;}

}

if(!flag) nums[result++] = nums[i];

}

return result;

}

}

1. Write a Java program to find the number of elements that is higher than the average of given array of integers.

public class Q20 {

public static int higherAvg(int[] nums) {

int sum = 0;

int counter = 0;

for (int num : nums)

sum += num;

double average = (double) sum / nums.length;

for (int num : nums)

if (num > average)

counter++;

return counter;

}

public static void main(String[] args) {

int [] nums = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int high = *higherAvg*(nums);

System.***out***.println("Nums higher than average: " + high);

}

}